CS 421: Computer Networks **FALL 2007**

MIDTERM II
December 4, 2007
120 minutes

Name: ______________________________

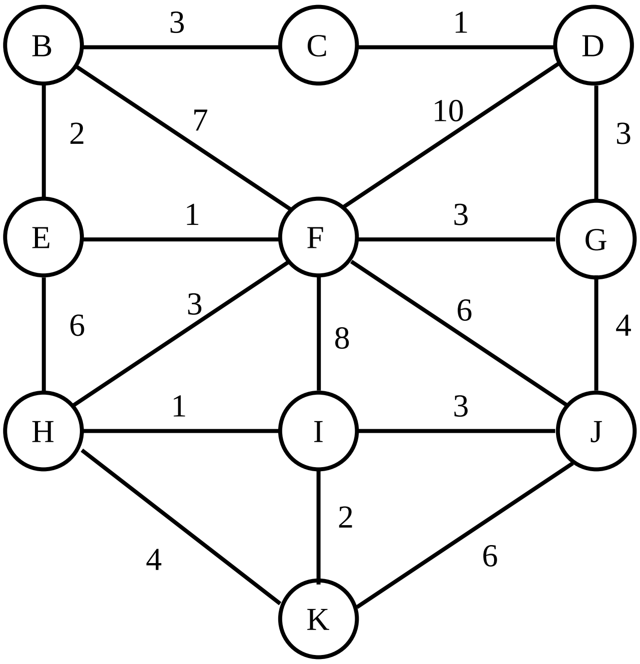
Student No: ______________________________

Show all your work very clearly. Partial credits will only be given if you carefully state your answer with a reasonable justification.

Q1	
Q2	
Q3	
TOT	

1)

a) (12 pts) Execute the Dijkstra algorithm **at node C** for the network shown below by filling in the following table. In the table, you need to give both the distance *D(v)* and the previous node *p(v)*.



iter.	*N*	*D(B), p(B)*	*D(D), p(D)*	*D(E), p(E)*	*D(F), p(F)*	*D(G), p(G)*	*D(H), p(H)*	*D(I), p(I)*	*D(J), p(J)*	*D(K), p(K)*

b) The network below uses the distance-vector routing algorithm. Assume the following:

- Links have the same cost in both directions.
- Nodes exchange their routing info once every second, in perfect synchrony and with negligible transmission delays. Specifically, at every t = k, k = 0, 1, 2, 3,..., each node sends and receives routing info instantaneously, and updates its routing table; the update is completed by time t=k+0.1.
- At time t = 0, the link costs are as shown below and the routing tables have been stabilized. At time t = 0.5, the link (4,5) becomes 10. There are no further changes in the link costs.
- Route advertisements are **only exchanged periodically**, i.e., there are no immediate route advertisements after a link cost change. Hence the first route advertisement after the link cost change at t = 0.5 occurs at t = 1.0. *Note:* However, whenever a link cost change occurs, two nodes at the endpoints of this link immediately make corresponding changes in their distance tables.

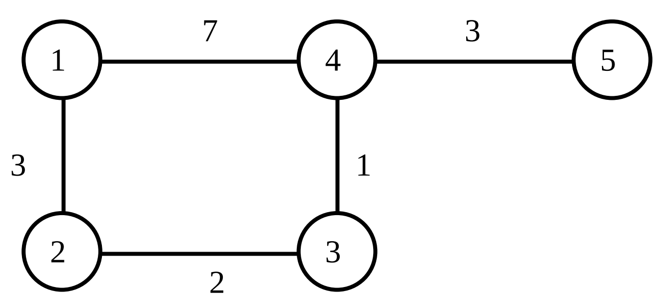
i) (12 pts) Assume that the distance vector algorithm **does not use poisoned reverse**. Give the evolution of the distance tables with respect to destination 5. Specifically, give the distance table entries for destination 5 at nodes 1-3, for t = 0.1, 0.5, 1.1, 2.1, ..., **until** all distance vectors stabilize. Present your final answer in the table given below where $D^i(j)$ is the distance vector denoting the distance from *i* to *j* .

Time, t	$D^1(5)$ via		$D^2(5)$ via		$D^3(5)$ via		$D^4(5)$ via		
	2	4	1	3	2	4	1	3	5
0.1									
1.1									
2.1									
3.1									
4.1									
5.1									
6.1									
7.1									
8.1									
9.1									
10.1									
11.1									
12.1									

ii) (12 pts) Redo part i. assuming that the distance vector algorithm **uses poisoned reverse**.

Time, t	$D^1(5)$ via		$D^2(5)$ via		$D^3(5)$ via		$D^4(5)$ via		
	2	4	1	3	2	4	1	3	5
0.1									
1.1									
2.1									
3.1									
4.1									
5.1									
6.1									
7.1									
8.1									

iii) (5 pts) Using the forwarding tables valid at t=3.5, find the paths followed by a packet sourced at node 1 and destined to node 5 for both parts i and ii above, i.e., with and without poisoned reverse.



2)

a) (6 pts) Name three methods/algorithms/protocols that make it possible to have more efficient use of IPv4 addresses.

b) (6 pts) Does counting-to-infinity problem occur in OSPF? Why or why not?

c) (8 pts) I decided to start a small company. I asked my ISP, myISP, to give me enough addresses for 1500 hosts. myISP owns the IP address block 151.18.0.0/16 and allocated a block from this address range and told me to use the following addresses (* can be any number in the range 0-255):

151.18.16.*
151.18.17.*
151.18.18.*
151.18.19.*
151.18.20.*
151.18.21.*

Since the sizes of the Internet routing tables have grown to huge proportions, I decided to announce the **fewest number of CIDR prefixes** to exactly cover all my company's IP addresses. Give the IP address block(s) that I need to use for advertising all addresses in my network, but no other addresses (use address/prefix format of CIDR).

b) You are given the assignment of setting subnet addresses for 4 buildings of your company. The number of Internet connected PCs in each building is given in the following table. Assume that the 148.64.64.0/18 address block is given to you for this purpose.

i) (8 pts) Use the following table to show the addresses of the four subnets that you have created.

Building	# of PCs	Subnet address (CIDR format)
1	3150	
2	2380	
3	1010	
4	880	

ii) (5 pts) What is the size of the **largest single** CIDR address block that you can assign from the unassigned addresses in the address block 148.64.64.0/18 remaining after you assigned the addresses to these four buildings?

3)

a) (4 pts) Consider a link state advertisement message exchanged between routers using OSPF as the intra-AS routing algorithm. What does the “upper layer protocol” field in the IPv4 header of the datagram corresponding to this message contain?

b) (4 pts) Consider a distance vector advertisement message exchanged between routers using RIP as the intra-AS routing algorithm. What does the “upper layer protocol” field in the IPv4 header of the datagram corresponding to this message contain?

c) (10 pts) Suppose that the TCP congestion window, CongWin, at a TCP sender is currently 8 KB and the slow start threshold, ssthresh, 10 KB (assume that 1KB=1000 Bytes). Assume that the maximum segment size, MSS, is 1000 Bytes. The sender sends 7 segments for a total of 7000 bytes of data. Assume that all segments except the 4^{th} segment are received successfully whereas 4^{th} segment is lost. Assume further that all ACKs are received successfully by the sender. What will be the value of CongWin and ssthresh in units of bytes after the processing of all ACKs is completed by the sender?

d) (8 pts) Assume a congestion feedback model for a system composed of two flows sharing a bottleneck link with bandwidth R bits/sec where both connections have the same RTT. Each flow gets binary synchronous feedback in discrete time steps of one RTT. If the aggregate consumption of the two flows is above the bottleneck bandwidth, both senders receive a congestion notification signal (CN), otherwise they receive no CN. The flows use a simple congestion control scheme: When no CN is received in a time step, each sender increases its window by one segment. On the other hand, for each congested time step, i.e., when both senders receive CN, each sender decreases its window by one segment. So, we can call this algorithm as Additive Increase-Additive Decrease (AIAD). **Prove or disprove** that AIAD achieves a fair allocation of bandwidth between the flows by using the following graph similar to the discussions we made in class in order to show that TCP’s AIMD algorithm is fair. Assume that the joint throughput vector is initially at point A.